

Implementation of EMC NetWorker and the NetWorker Module for Oracle with RAC on Oracle 9i and Red Hat GFS

Applied Technology

Abstract

EMC completed a proof of concept to qualify NetWorker[®] on the Red Hat Global File System (GFS). This white paper outlines the configuration for this test environment and describes the NetWorker configuration used to provide an end-to-end solution for “hot” online backup of an Oracle database in a Real Application Cluster environment installed on Red Hat GFS 6.0.

May 2007

Copyright © 2007 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com

All other trademarks used herein are the property of their respective owners.

Part Number H2794

Table of Contents

Executive summary	4
Introduction	4
Audience	4
Terminology	4
Red Hat Global File System	5
Oracle Real Application Cluster	5
Advantages	6
EMC NetWorker backup restore architecture.....	6
EMC NetWorker Module for Oracle backup restore architecture	7
Oracle backup and recovery system	7
Oracle Recovery Manager	7
How NMO backs up data	8
How NMO restores data	9
Configuration.....	10
Test environment.....	11
Prerequisites	11
Configuring GFS.....	12
Installation	12
Configuring the pools	12
Creating the GFS configuration files	14
Creating the Cluster Configuration Archive (CCA).....	15
Creating and mounting GFS.....	15
Installing and configuring Oracle 9i RAC on Red Hat GFS	15
Modifying \$ORACLE_HOME/oracm/admin/cmcfg.ora	16
Loading the hangcheck-timer module on all nodes.....	16
Enable oracm CDPN	16
Starting Oracle Cluster Manager.....	16
Install Oracle RDBMS	16
Creating network configuration files for Oracle instance.....	17
Enabling network CDPN.....	18
Starting the listener	18
Setting up the cluster configuration files	18
Configuration of the NMO client in RAC environment for backup	19
Sample RMAN backup scripts.....	19
Sample RMAN restore scripts	19
Conclusion	19
References	20

Executive summary

In the second quarter of 2006, there was a requirement from multiple customers to support backup of data on Red Hat Global File System 6.0 with EMC® NetWorker® for applications such as Oracle and Sendmail that reside on GFS.

EMC conducted a feasibility analysis and certified that NetWorker can back up GFS, which made possible the qualification of NetWorker Modules, specifically the NetWorker Module for Oracle, to back up database objects residing on GFS. This effort helps ensure that the NetWorker solution can be recommended for GFS support requirements.

Introduction

This white paper provides insight into how to configure Oracle Real Application Cluster (RAC) on Red Hat GFS and use NetWorker functionality to back up the Oracle database running in this configuration.

An Oracle database is a collection of data managed by an Oracle database management system or DBMS. This is one of the most widely used databases running across different platforms. Linux is growing as one of the common environments to run the Oracle database, and one of the most common configurations is Oracle running in a RAC environment with multiple nodes for high availability.

NetWorker is EMC's flagship product for backup and restore of enterprise data, which can range from simple file system backup to application backups of Oracle. To back up each application, NetWorker provides plug-ins called NetWorker Modules. The NetWorker Module for Oracle implements an Oracle System Backup to Tape (SBT) interface to take backups of Oracle databases.

Red Hat Global File System provides scalable performance and capacity for Red Hat Enterprise Linux clusters. By using storage area networking and sophisticated clustering techniques provided by GFS and Red Hat Cluster Manager, system administrators can create large Red Hat Enterprise Linux clusters for the most demanding enterprise applications.

Audience

This white paper can be used by anyone configuring an Oracle 9i RAC environment on the Red Hat GFS and using NetWorker to back up the database. This paper discusses how the NetWorker Module for Oracle backs up the database in hot backup (ARCHIVELOG) mode. Basic understanding of the Oracle SBT interface and Oracle Recovery Manager (RMAN) is a must to understand the implementation.

Terminology

NetWorker Module for Oracle (NMO): A plug-in for a NetWorker client that backs up and restores Oracle database objects.

Oracle Recovery Manager (RMAN): A utility that is used for backup and restore of Oracle database objects.

Red Hat Global File System (GFS): Helps to share data across multiple cluster nodes with a common pool of storage.

System Backup to Tape (SBT): Interface to back up and restore Oracle objects. Third-party backup vendors implement this interface that integrates Oracle backup and restore functionality with media management functionality of backup solutions.

Red Hat Global File System

Red Hat GFS is a cluster file system that provides data sharing among Linux-based computers. GFS provides a single, consistent view of the file system namespace across all nodes in a cluster, and it allows applications to install and run without much knowledge of the underlying storage infrastructure. GFS is fully compliant with the IEEE POSIX interface, allowing applications to perform file operations as if they were running on a local file system. Also, GFS provides features that are typically required in enterprise environments, such as quotas, multiple journals, and multipath support.

GFS provides a versatile method of networking your storage according to the performance, scalability, and economic needs of your storage environment. Figure 1 shows GFS in a SAN architecture.

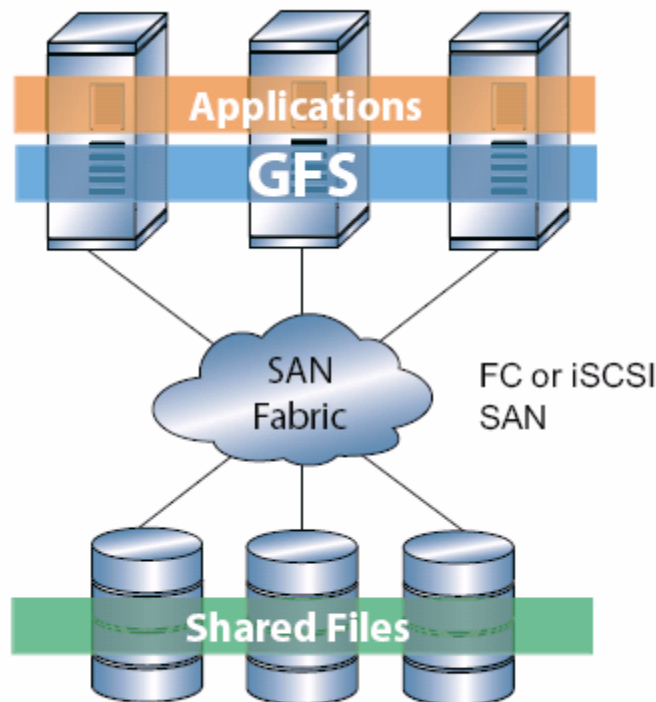


Figure 1. GFS with a SAN¹

Oracle Real Application Cluster

Real Application Cluster (RAC) is a clustered database. Oracle came up with the RAC concept as an extension of cluster technology. Clusters are a group of resources that provide fault tolerance against a single point of failure.

The RAC architecture segregates the physical storage (datafiles are the physical entity; tablespaces and tables are logical entities) required for the database with the memory/CPU requirements. The concept is to build up a database on shared storage that can be accessed by the multiple instances (nodes) running on different physical hosts.

¹ *Red Hat GFS 6.0 Administrator's Guide*

Advantages

- **Scalable:** Easily scalable, and whenever the need arises, more nodes can be added to the existing cluster.
- **No single point of failure:** Each instance running in the cluster has the same level of access to the database. This provides fault tolerance against a single point of failure.
- **Load balancing:** As each instance has a parallel and same level of access to the database, it provides load balancing at a lower total cost of ownership.
- **Inherits Oracle features:** All Oracle features like security, scalability, management, and robustness were inherited in the RAC environment.

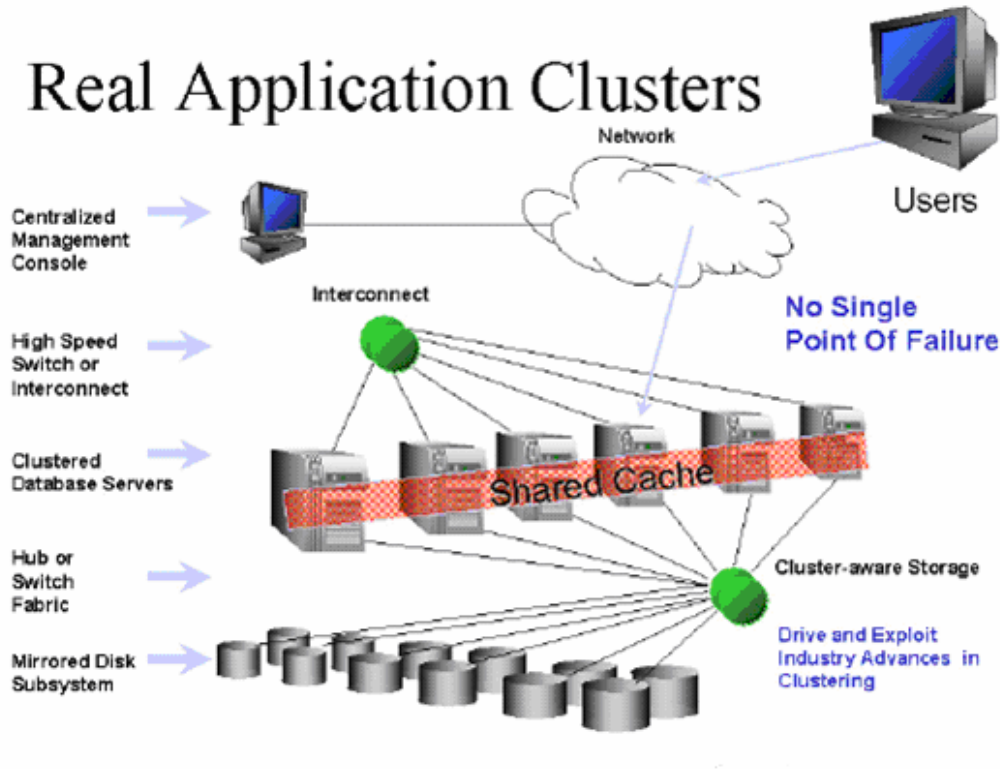


Figure 2. RAC architecture²

EMC NetWorker backup restore architecture

EMC NetWorker utilizes a client/server architecture comprising of three distinct functions within a data zone – client, storage nodes, and server – to provide the capabilities and performance required to protect and manage data on the most complex networks. NetWorker enables administrators to centrally manage and automate backup operations within a data zone, while applying varying storage management policies for individual clients.

There are different NetWorker Modules that back up specific applications like SQL, Oracle, SAP, and so on. These modules are plug-ins for the NetWorker client which, when coupled with NetWorker client software, integrate the NetWorker solution with specific applications. These modules integrate NetWorker functionality by implementing the interfaces and APIs exposed to the third-party backup vendors by the underlying applications.

² <http://www.oracle.com>

EMC NetWorker Module for Oracle backup restore architecture

The NetWorker Module for Oracle (NMO) software is an add-on module for NetWorker that enables you to perform the following tasks:

- Manual Oracle backups
- Scheduled Oracle backups
- Restores of Oracle backup data
- Automated media management

NMO provides the following features:

- Capability to integrate database and file system backups, to relieve the burden of backup from the database administrator while allowing the administrator to retain control of the restore process
- Automatic database storage management through automated scheduling, autochanger support, electronic tape labeling, and tracking
- Support for backup to a centralized backup server
- High performance through support for multiple, concurrent high-speed devices such as a super digital linear tape (SDLT) drive

Together with the NetWorker server, NMO augments the backup and recovery system provided by the Oracle Server and provides a storage management solution that addresses the need for cross-platform support of enterprise applications.

Oracle backup and recovery system

The NMO and NetWorker server and client software function with the standard Oracle backup and recovery system to produce an efficient Oracle data storage management system. The implementation of an Oracle backup and recovery strategy requires knowledge of how NetWorker software works with the Oracle components.

The regular Oracle backup and recovery system consists of the following:

- Oracle Server
- RMAN
- RMAN recovery catalog (optional)
- Backup management tools for Oracle Enterprise Manager (OEM) (optional)

The NetWorker software consists of the following components:

- NetWorker server
- NetWorker client
- NMO

Oracle Recovery Manager

The MediaManagement/ProxyCopy API is also referred to as the System Backup to Tape (SBT) API. It defines the interface between Oracle Recovery Manager (RMAN) and third-party backup software packages.

During Oracle backup and restore operations, NMO acts as an intermediary between the NetWorker server and RMAN. RMAN is the main Oracle utility for backing up, restoring, and recovering Oracle datafiles,

control files, and archived redo log files. It stores information about its operations in the control file of the backed-up database (the target database) and, optionally, in the recovery catalog on the Oracle Server host.

RMAN provides the following features:

- Online and offline Oracle database backups
- High performance through parallel backups and restores
- An intelligent interface to Oracle databases
- Well-defined backup strategies
- Checks for corrupt blocks during backups and restores
- Block-level incremental backups and restores

How NMO backs up data

To provide storage management services for Oracle Server data, NMO implements the media management interface, also known as the Oracle System Backup to Tape (SBT). This interface comprises a media management library (MML) that is integrated with the Oracle Server during the NMO installation on the Oracle Server host.

The module installation links the MML with the Oracle kernel software. This enables the Oracle Server processes to call the MML software routines to back up and restore Oracle data to and from the media controlled by the NetWorker server.

A regular Oracle backup can be performed in one of two ways:

- By issuing the appropriate commands through the RMAN command line interface
- By using the OEM Backup Management Tools, which include an optional graphical user interface to the RMAN utility

RMAN establishes connections with the target database. Each connection starts an Oracle Server process that performs the backup. During the backup, the following occurs:

1. The Oracle Server processes read the datafile, control file, or archived redo log being backed up, then write it to the NetWorker server through the MML.
2. The NetWorker server stores the Oracle data to the appropriate backup storage device.
3. At the end of the Oracle backup, the NetWorker server updates the online client and media indexes with information about the backup media used and the Oracle data it contains.

A manual Oracle backup is initiated by Oracle RMAN or Oracle Enterprise Manager (OEM). NMO considers a backup scheduled through OEM as a manual backup. When the RMAN utility is invoked for a backup, Oracle Server processes call the NMO media management library software routines to initiate the backup. A regular manual Oracle backup includes the following interactions:

1. The Oracle Server processes (one for each allocated channel) contact the nsrexecd service to obtain the NetWorker client information.
2. The Oracle Server processes contact the main NetWorker service, nsrd, to obtain the required authorization and the identifier of the nsrmmmd process that manages the requested backup device.
3. The Oracle Server processes send the backup data to the NetWorker media service, nsrmmmd, to store on the appropriate backup volumes.
4. Tracking information is stored in the NetWorker online indexes:

- a. The nsrmmd service records tracking information in the NetWorker media database by using the nsrmmdbd service.
- b. The Oracle processes send tracking information to the NetWorker client file index by using the nsrindexd service.

Figure 3 shows the Oracle backup process performed by the NetWorker Module for Oracle and NetWorker.

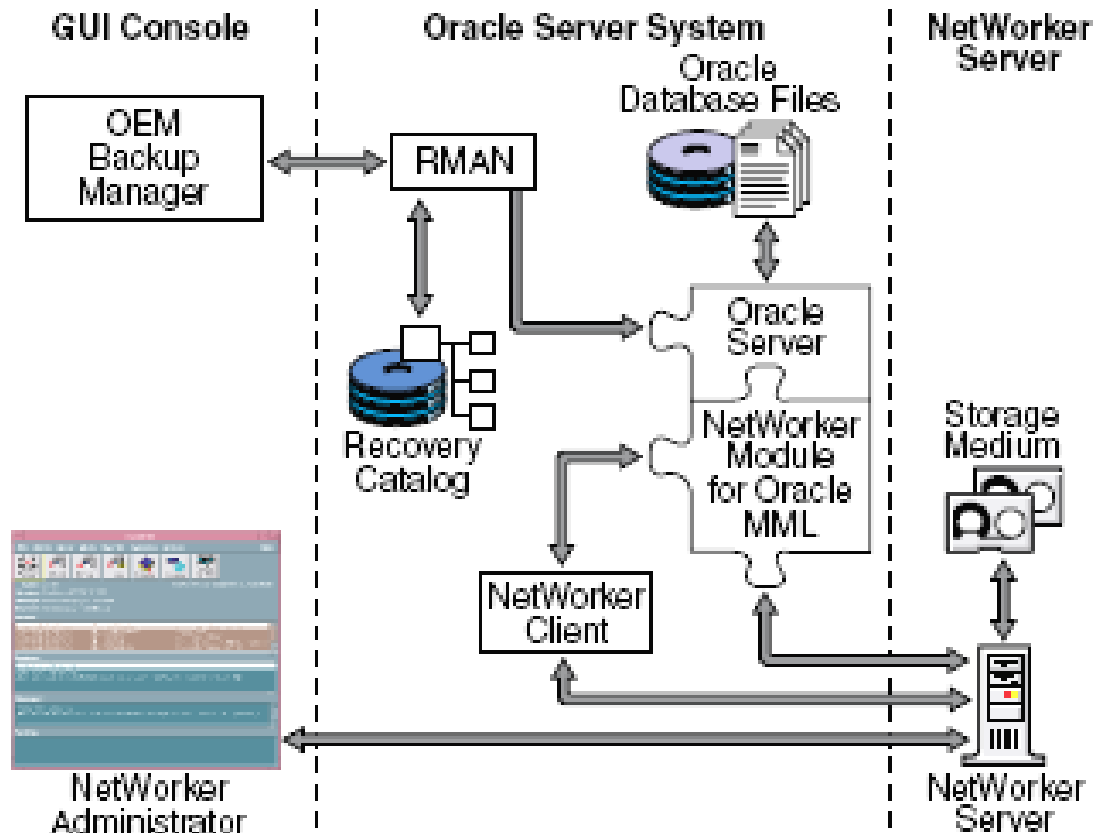


Figure 3. NMO backup process³

How NMO restores data

An Oracle restore can be performed in one of two ways:

- By issuing the appropriate commands through the RMAN command line interface
- By using the OEM backup management tools, which include an optional graphical user interface to the RMAN utility

The RMAN utility starts Oracle Server processes on the target database. These Oracle Server processes initiate the restore by calling NMO media management library software routines. A regular Oracle data restore includes the following interactions:

1. The MML translates the object names requested by RMAN into a format that the NetWorker server understands and then forwards the names to the NetWorker service, nsrindexd.

³ *NetWorker Module for Oracle Release 4.2 Multiplatform Version Administrator's Guide*

2. The nsrindexd service verifies that the backup pieces exist in the client file index.
3. When the NetWorker server receives a restore request from the client, the main NetWorker service, nsrd, contacts the media service, nsrmmmd.
4. The nsrmmmd service calls the media database service, nsrmmdbd, to determine which media contains the requested save set and issue a mount request.
5. The nsrmmmd service reads the appropriate backup volumes and passes the data to the Oracle processes through the NMO media management library.
6. The Oracle Server processes write the data to disk.

Figure 4 shows the Oracle restore process performed by the NetWorker Module for Oracle and NetWorker.

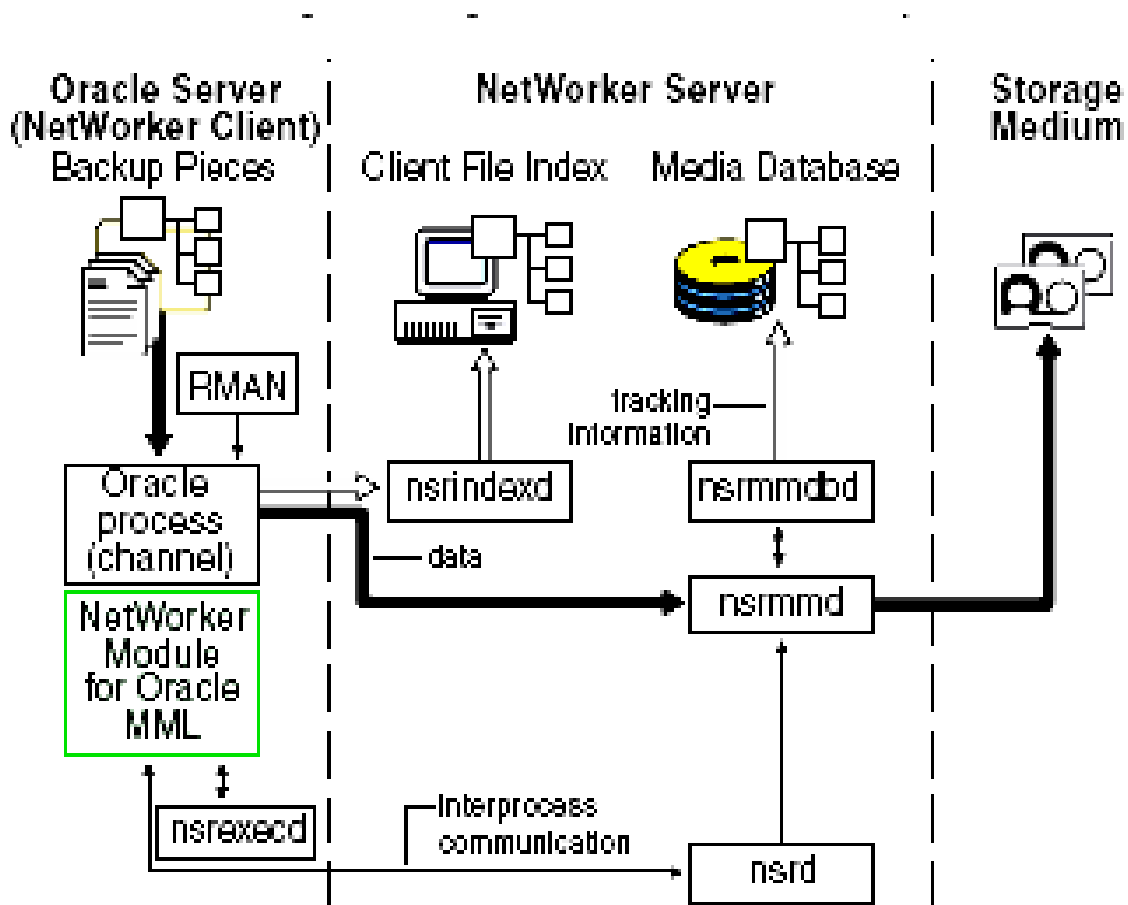


Figure 4. NMO restore proces⁴

Configuration

This section will walk you through the steps to configure RAC on Red Hat GFS. This work is done to qualify NetWorker and the NetWorker Module for Oracle 4.2 to back up and restore Oracle database

⁴ *NetWorker Module for Oracle Release 4.2 Multiplatform Version Administrator's Guide*

objects seamlessly. With this qualification, customers can implement the NetWorker backup and restore solution for their data existing on the Red Hat GFS.

Test environment

Hardware configuration

- Dell 2850
- HBA: Emulex 90000 and QLogic 2342
- Red Hat Enterprise Linux ES release 3 (Taroon update 5) (kernel 2.4.21-32.ELsmp)
- CLARiiON® CX700 running with FLARE® 22 for importing LUNs to create GFS
- Two-node cluster

Software configuration

- NetWorker 7.3
- Oracle 9.2.0.0
- NetWorker Module for Oracle v4.2
- EMC PowerPath® 4.3.2-011
- GFS 6.0

Table 1 shows the network node names and IP addresses.

Table 1. Network nodes and IP addresses

Public network		Private network	
Node name	IP address	Node name	IP address
rmdevlin1	10.31.79.166	rmnode1	192.168.1.1
rmdevlin2	10.31.79.167	rmnode2	192.168.1.2

Prerequisites

The following are the mandatory items to check on your system test environment:

- Verify that hardware is compatible as mentioned in *EMC Support Matrix*.
- Verify that the HBAs are supported on OS and EMC hardware (CLARiiON in this case).
- Proper zoning: There should be one zone having a CLARiiON Port WWN and Host HBA port WWN. Don't create zone with multiple hosts.
- Once LUNs are exported to the hosts, verify the Serial Number for these LUNs. EMC has used a custom script. But any script that returns a Serial Number of the device can be used to verify the disks.
- Hosts should be able to communicate with one another properly. Verify this using nslookup and the ping command.
- Verify that the legacy network packages and development tools are installed on both hosts.
- Make sure the hosts have all the required Oracle patches (including kernel parameters and space constraints).
- Make sure both cluster nodes have the following packages installed:
 - Perl-Net-Telnet module (on the operating system CD)
 - NTP services (Specify the NTP server while installing the operating system)
- Make sure that rsh is enabled between two hosts. Configure the .rhosts and /etc/hosts.allow file.

```

-----Sample hosts.allow-----
-----
[root@rmdevlin1 etc]# more hosts.allow
#
# hosts.allow    This file describes the names of the hosts which are
#                allowed to use the local INET services, as decided
#                by the '/usr/sbin/tcpd' server.
#
rmdevlin1
rmdevlin2
rmnode1
rmnode2

```

```

-----Sample .rhosts-----
-----
[root@rmdevlin1 root]# more .rhosts
rmdevlin1.lss.emc.com
rmdevlin2.lss.emc.com
rmnode1.lss.emc.com
rmnode2.lss.emc.com
rmnode1
rmnode2
rmdevlin1
rmdevlin2

```

Configuring GFS

Before configuring GFS on Red Hat Linux, you should be able to answer the following questions:

- How many LUNS (number and size) do you need from the storage array?
- How many file systems do you want to create?
- How will these file systems be distributed on each disk? (In case you want to create more than one file system on one disk.)

Installation

Install the GFS software, which you can download from Red Hat (<http://www.redhat.com>). The GFS user tools and GFS kernel tools must be installed in the same order.

Once the GFS RPMs are installed, load the following kernel modules:

- pool.o
- lock_gulm.o
- gfs.o

The following is the method and order to load these modules:

1. depmod -a
2. modprobe pool
3. modprobe lock_gulm
4. modprobe gfs

Run the lsmod command to validate that all the modules are loaded.

Configuring the pools

We have to configure the volume layout, called pools, that will be used in our cluster configuration. Table 2 shows the pool layout for the configuration used. It's always better to have the configuration of the pool prepared before actually creating these pools.

Table 2. Pool layout

Pool no.	Pool name	Purpose	Size of the slice/disk
1	Oraraw1	Oracle cluster software	100-200 MB /dev/sda1
2	Oraraw2	Server control software	100-200 MB /dev/sda2
3	Ccs_archive	Cluster configuration system	200-500 MB /dev/sda3
4	Oracle_base	Oracle_Home directory	8-10 GB /dev/sdb1
5	Oraundo	Undo Tablespace datafiles	5-10 GB /dev/sdc1
6	Oraindex	Index Tablespace datafiles	5-10 GB /dev/sdd1
7	Oradata	For System / User Tablespace datafiles	10-20 GB /dev/sde1

Use the example below to configure your pool:

```
poolname <Pool Name>
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 <Slice name>
```

To create the Oradata.pool, use the following:

```
poolname Oradata
subpools 1
subpool 0 0 1 gfs_data
pooldevice 0 0 /dev/sde1
```

Run the pool_tool command for all pools created from any of the cluster nodes:

```
rmdevlin1.lss.emc.com # pool_tool -c Oraraw1.pool
Pool label written successfully from Oraraw1.pool
rmdevlin1.lss.emc.com # pool_tool -c Oraraw2.pool
Pool label written successfully from Oraraw2.pool
rmdevlin1.lss.emc.com # pool_tool -c Ccs_archive.pool
Pool label written successfully from Ccs_archive.pool
rmdevlin1.lss.emc.com # pool_tool -c Oracle_base.pool
Pool label written successfully from Oracle_base.pool
rmdevlin1.lss.emc.com # pool_tool -c Oraundo.pool
Pool label written successfully from Oraundo.pool
rmdevlin1.lss.emc.com # pool_tool -c Oraindex.pool
Pool label written successfully from Oraindex.pool
rmdevlin1.lss.emc.com # pool_tool -c Oradata.pool
Pool label written successfully from Oradata.pool
```

In order to make these pools available from each cluster, run the pool_assemble command from each node:

```
rmdevlin1.lss.emc.com #pool_assemble
Oraraw1.pool assembled
Oraraw2.pool assembled
Ccs_archive.pool assembled
Oracle_base.pool assembled
Oraundo.pool assembled
Oraindex.pool assembled
Oradata.pool assembled

rmdevlin2.lss.emc.com #pool_assemble
Oraraw1.pool assembled
Oraraw2.pool assembled
Ccs_archive.pool assembled
Oracle_base.pool assembled
Oraundo.pool assembled
Oraindex.pool assembled
Oradata.pool assembled
```

Creating the GFS configuration files

You have to create three GFS configuration files – cluster.ccs, fence.ccs, and nodes.ccs.

Cluster.ccs

The cluster file contains the name of the cluster and cluster nodes where LOCK_GULM needs to be run. A sample file is shown below:

```
cluster {
    name = "gfsrac"
    lock_gulm {
        servers = ["rmdevlin1", " rmdevlin2"]
    }
}
```

cluster.ccs

Fence.ccs

The fence.ccs file describes each device used for fencing. Fencing defines the device that will be used for fault tolerance for various components in the cluster configuration. If a node has dual power supplies, you must define a fencing device for each power supply. Similarly, if a node has multiple paths to Fibre Channel storage, you must define a fencing device for each path to the FC storage. All the fencing devices intended to be used in the cluster configuration have to be defined in fence.ccs. The *Red Hat GFS 6.0 Administrator's Guide* provides the syntax for defining various fencing devices. A sample file is shown below:

```
fence_devices {
    manual 1 {
        agent = fence_manual
    }
}
```

fence.ccs

Nodes.ccs

This file has an entry for each node and the LOCK_GULM server node. It contains the fencing parameter and the IP address for each node. A sample file is shown below

```
nodes {
    rmnode1 {
        ip_interfaces {
            eth0 = "192.168.1.1"
        }
        fence {
            power {
                manual1
            }
        }
    }

    rmnode2 {
        ip_interfaces {
            eth0 = "192.168.1.2"
        }
        fence {
            power {
                manual1
            }
        }
    }
}
```

nodes.ccs

Creating the Cluster Configuration Archive (CCA)

You have to create the CCA file on the shared storage. The file can be created from any of the participating nodes. Put all the configuration files created in a directory (say /root/gfs/temp) and run the following command from any one node:

```
rmdevlin1.lss.emc.com # ccs_tool create /root/gfs/temp /dev/pool/ccs_archive
```

Start the CCS daemon from all the nodes:

```
rmdevlin1.lss.emc.com# ccsd -d /dev/pool/ccs_archive
rmdevlin2.lss.emc.com# ccsd -d /dev/pool/ccs_archive
```

Start the LOCK servers:

```
rmdevlin1.lss.emc.com# lock_gulmd
rmdevlin2.lss.emc.com# lock_gulmd
```

Creating and mounting GFS

We have already configured the pools Oracle_base, Oraundo, Oraindex, Oradata, Oraraw1, Oraraw2, and Ccs_archive. Now configure the GFS on all these pools that will then be used for storing the Oracle data. Run the following command from any one of the participating nodes in the cluster:

```
rmdevlin1.lss.emc.com# gfs_mkfs -p lock_gulm -t gfsrac:Oracle_base -j 4
/dev/pool/Oracle_base
rmdevlin1.lss.emc.com# gfs_mkfs -p lock_gulm -t gfsrac:Oraundo -j 4 /dev/pool/ Oraundo
rmdevlin1.lss.emc.com# gfs_mkfs -p lock_gulm -t gfsrac:Oraindex -j 4 /dev/pool/ Oraindex
rmdevlin1.lss.emc.com# gfs_mkfs -p lock_gulm -t gfsrac:Oradata -j 4 /dev/pool/ Oradata
```

Create the mount points on each node that will mount these file systems:

```
rmdevlin1.lss.emc.com# mkdir -p /mnt/Oracle_base
rmdevlin1.lss.emc.com# mkdir -p /mnt/Oraundo
rmdevlin1.lss.emc.com# mkdir -p /mnt/Oraindex
rmdevlin1.lss.emc.com# mkdir -p /mnt/Oradata
```

Mount the file systems at the mount point created on each node:

```
rmdevlin1.lss.emc.com# mount -t gfs /dev/pool/Oracle_base /mnt/Oracle_base
rmdevlin1.lss.emc.com# mount -t gfs /dev/pool/Oraundo /mnt/Oraundo
rmdevlin1.lss.emc.com# mount -t gfs /dev/pool/Oraindex /mnt/Oraindex
rmdevlin1.lss.emc.com# mount -t gfs /dev/pool/Oradata /mnt/Oradata
```

Set up the raw devices (mandatory for Oracle 9i RAC). Bind the raw devices by editing the /etc/sysconfig/rawdevices file on each node:

```
rmdevlin1.lss.emc.com#more /etc/sysconfig/rawdevices
/dev/raw/raw1 /dev/pool/Oraraw1
/dev/raw/raw2 /dev/pool/Oraraw2

rmdevlin1.lss.emc.com#service rawdevices restart
```

You can validate the raw devices' binding and status by running the commands raw -qa and chkconfig -list rawdevices.

Installing and configuring Oracle 9i RAC on Red Hat GFS

Follow the standard Oracle Database Installation Guide for:

- Prerequisites for hardware configurations
- Kernel parameters
- Compilers needed
- Oracle users and groups (oracle and dba are used in this paper)

Give the ownership of the following mount points to oracle:dba:

```
rmdevlin1.lss.emc.com# chown -R oracle:dba /mnt/Oracle_base
rmdevlin1.lss.emc.com# chown -R oracle:dba /mnt/Oraundo
rmdevlin1.lss.emc.com# chown -R oracle:dba /mnt/Oraindex
```

Implementation of EMC NetWorker and the NetWorker Module for Oracle with RAC on Oracle 9i and Red Hat GFS

```
rmdevlin1.lss.emc.com# chown -R oracle:dba /mnt/Oradata
rmdevlin1.lss.emc.com# chown -R oracle:dba /dev/raw/raw1
rmdevlin1.lss.emc.com# chown -R oracle:dba /dev/raw/raw2
```

Install Oracle Cluster Manager and provide the inputs specific to your environment:

Oracle Base Directory: Database installation will be done.

Oracle Home: Equivalent to your \$ORACLE_HOME environment variable.

Choose **Oracle Cluster Manager** in the list of available products.

Public Nodes: Hostname of all the nodes (rmdevlin1.lss.emc.com and rmdevlin2.lss.emc.com for our setup).

Private Nodes: Private node hostname (rmnode1 and rmnode2 for our setup).

Quorum: Raw device mapped to oraraw1 (/dev/raw/raw1 for our setup).

Modifying \$ORACLE_HOME/oracm/admin/cmcfgr.ora

*** Sample File ***

```
###cmcfgr.ora###
HeartBeat = 20000
ClusterName = Oracle Cluster Manager, version 9i
PollInterval = 1000
MissCount = 300
PrivateNodeNames=rmnode1,rmnode2
PublicNodeNames=rmdevlin1,rmdevlin2
ServicePort=9989
KernelModuleName=hangcheck-timer
cmDiskFile=/dev/raw/raw1
HostName=rmnode1
```

Loading the hangcheck-timer module on all nodes

```
rmdevlin1.lss.emc.com# insmod hangcheck-timer hangcheck_tick=30 hangcheck_timer=180
rmdevlin2.lss.emc.com# insmod hangcheck-timer hangcheck_tick=30 hangcheck_timer=180
```

Enable oracm CDPN

Follow the steps on any one node:

```
rmdevlin1.lss.emc.com#cd $ORACLE_HOME
rmdevlin1.lss.emc.com#mkdir rmdevlin1 rmdevlin2
rmdevlin1.lss.emc.com#su - oracle
rmdevlin1.lss.emc.com$ for i in `ls -d rmdevlin*`; do cp -r oracm $i; done
rmdevlin1.lss.emc.com$ mv oracm oracm.original
rmdevlin1.lss.emc.com$ ln -s @hostname/oracm oracm
```

Verify from each hostname that the file cmcfgr.ora has the right value for the **HostName** parameter or manually change it specific to your setup

Starting Oracle Cluster Manager

Run the following steps on each node with the root user:

```
rmdevlin1.lss.emc.com# . ~oracle/.bash_profile
rmdevlin1.lss.emc.com# cd $ORACLE_HOME/oracm/bin
rmdevlin1.lss.emc.com# .ocmstart.sh
```

Install Oracle RDBMS

Create the directory /var/opt/oracle and give the Oracle user ownership of this directory (chown -R oracle:dba /var/opt/oracle). During installation, provide inputs specific to your setups:

Available products: Oracle 9i Database.

Choose **Custom Installation** and select Oracle 9i Real Application Cluster from the available product components.

Shared Configuration File: Points to the raw device mapped to Oraraw2 (/dev/raw/raw2).

When installation is done, validate for the file /var/opt/oracle/srvConfig.loc.

Creating network configuration files for Oracle instance

You have to configure the listener.ora, tnsnames.ora and sqlnet.ora files specific to your setups for both nodes:

```
### tnsnames.ora ###
# LISTENER.ORA.RMDEVLIN1 Network Configuration File:
/mnt/oracle_base/product/920/network/admin/listener.ora.rmdevlin1
# Generated by Oracle configuration tools.

LISTENERS_RAC1.lss.emc.com =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin1) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin2) (PORT = 1521))
  )

LISTENER_RAC12.lss.emc.com =
  (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin2) (PORT = 1521))

LISTENER_RAC11.lss.emc.com =
  (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin1) (PORT = 1521))

RAC12.lss.emc.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin2) (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = rac1)
      (INSTANCE_NAME = rac12)
    )
  )

RAC11.lss.emc.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin1) (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = rac1)
      (INSTANCE_NAME = rac11)
    )
  )

### listener.ora ###

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = rmdevlin1) (PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /mnt/oracle_base/product/920)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (ORACLE_HOME = /mnt/oracle_base/product/920)
      (SID_NAME = rac11)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = rac1)
      (ORACLE_HOME = /mnt/oracle_base/product/920)
      (SID_NAME = rac1)
    )
  )
```

```
)
)

### sqlnet.ora ###

NAMES.DEFAULT_DOMAIN = lss.emc.com
NAMES.DIRECTORY_PATH= (TNSNAMES)
```

Enabling network CDPN

```
rmdevlin1.lss.emc.com#cd $ORACLE_HOME
rmdevlin1.lss.emc.com#mkdir rmdevlin1 rmdevlin2
rmdevlin1.lss.emc.com#su - oracle
rmdevlin1.lss.emc.com$ for i in `ls -d rmdevlin*`; do cp -r network $i; done
rmdevlin1.lss.emc.com$ mv network network.original
rmdevlin1.lss.emc.com$ ln -s @hostname/network network
```

Verify from each hostname that all configuration files under the network directory reflect the node-specific information correctly.

Starting the listener

Start the listener using `lsnrctl start <Listener>`.

Setting up the cluster configuration files

Execute the following commands from any node of the cluster with the Oracle user:

```
rmdevlin1.lss.emc.com$ srvconfig -init
rmdevlin1.lss.emc.com$ gsdctl start
rmdevlin1.lss.emc.com$ srvctl -add instance -d <Instance you created> -o $ORACLE_HOME
rmdevlin1.lss.emc.com$ srvctl -add instance -d <Instance you created> -i Instancename1
rmdevlin1
rmdevlin1.lss.emc.com$ srvctl -add instance -d <Instance you created> -i Instancename2
rmdevlin2
```

From each node run the following command:

```
rmdevlin1.lss.emc.com$ gsdctl start
rmdevlin1.lss.emc.com$ agentctl start
```

You should now be able to run your database with RAC configured. If you face any issue while bringing up the instance, check the init.ora file for the instance and modify if required. The sample init.ora for this setup follows:

```
cluster_database_instances=2
cluster_database=true
rac11.instance_name=rac11
rac11.instance_number=1
rac11.local_listener=LISTENER_RAC11
#rac11.local_listener=RAC1.LSS.EMC.COM
rac11.thread=1
#rac11.undo_tablespace=UNDOTBS1
rac12.instance_name=rac12
rac12.instance_number=2
rac12.local_listener=LISTENER_RAC12
#rac12.local_listener=RAC1.lss.emc.com
rac12.thread=2
#rac12.undo_tablespace=UNDOTBS2
background_dump_dest='/mnt/oracle_base/admin/rac1/bdump'
cluster_database=true
compatible='9.2.0.0.0'
control_files='/mnt/oracle_base/oradata/rac1/control01.ctl','/mnt/oracle_base/oradata/rac1/control02.ctl','/mnt/oracle_base/oradata/rac1/control03.ctl'
core_dump_dest='/mnt/oracle_base/admin/rac1/cdump'
db_block_size=8192
db_cache_size=25165824
db_domain=''
db_file_multiblock_read_count=16
db_name='rac1'
```

Implementation of EMC NetWorker and the NetWorker Module for Oracle with RAC on Oracle 9i and Red Hat GFS

```

dispatchers='(PROTOCOL=TCP) (SERVICE=rac1XDB)'
fast_start_mttr_target=300
hash_join_enabled=TRUE
java_pool_size=83886080
job_queue_processes=10
large_pool_size=8388608
open_cursors=300
pga_aggregate_target=25165824
processes=150
query_rewrite_enabled='FALSE'
remote_listener='LISTENERS_RAC1'
remote_login_passwordfile='exclusive'
shared_pool_size=83886080
sort_area_size=524288
star_transformation_enabled='FALSE'
timed_statistics=TRUE
undo_management='AUTO'
undo_retention=10800
user_dump_dest='/mnt/oracle_base/admin/rac1/udump'

```

Configuration of the NMO client in a RAC environment for backup

- Install NMO on *each* node of the cluster to be used for backup and recover operations.
- Configure a client resource on the NetWorker server for the virtual host and each physical host that will run backups and restores. In each client resource, use the fully qualified IP name as the client name.
- In the Remote Access attribute in the client resource for a virtual host, specify the name of each physical host that can store and retrieve the backups.
- Create the appropriate RMAN backup and restore scripts for the preferred types of Oracle backups and restores on the RAC system.

Sample RMAN backup scripts

```

connect target sys/oracle@rac11;
run {
  allocate channel t_1 type SBT_TAPE
  parms
'ENV=(NSR_DEBUG_LEVEL=9,NSR_PS_DEBUG_LEVEL=9,NSR_DEBUG_FILE=/nsr/applogs/shelesh1.
log)';
  send 'NSR_ENV=(NSR_DATA_VOLUME_POOL=Default,NSR_SERVER=rmdevlin1)';
  backup full tablespace 'TS Name';
  release channel t_1;
}

```

Sample RMAN restore scripts

```

connect target sys/nmo;
run
{
  allocate channel t_2 type SBT_TAPE
  parms 'ENV=(NSR_SERVER=rmdevlin1,NSR_DEBUG_LEVEL=9,NSR_PS_DEBUG_LEVEL=9)';
  send 'NSR_ENV=(NSR_SERVER=rmdevlin1,NSR_DATA_VOLUME_POOL=Default)';
  restore (tablespace <TS NAME>);
  recover tablespace <TS NAME>;
  sql 'alter tablespace <TS NAME> online';
  release channel t_2;
}

```

Conclusion

NetWorker and the NetWorker Module for Oracle are now fully qualified to back up and restore Oracle database objects on Red Hat GFS 6.0. Customers who have mission-critical Oracle data running in these types of environments can now use the EMC flagship product for backup and recovery and can leverage the EMC Information Lifecycle Management (ILM) strategy for a complete enterprise solution for their data at all stages in its life cycle.

References

EMC resources are listed below. For more information, go to [EMC.com](http://www.emc.com).

- *EMC NetWorker Release 7.3.2 Multiplatform Edition Administration Guide*
- *EMC NetWorker Release 7.3.2 Linux Version Installation Guide*
- *EMC NetWorker Module for Oracle Release 4.2 Multiplatform Version Administrator's Guide*
- *EMC NetWorker Module for Oracle Release 4.1 Multiplatform Version Installation Guide*

For more information on Oracle, go to its website: <http://www.oracle.com>

For more information on Red Hat Global File System, go to its website (<http://www.redhat.com/software/rha/gfs/>) and consult the following guide:

- *Red Hat GFS 6.0 Administrator's Guide*